جامعة الملك سعود
King Saud University

# GE 211

# C++ Programming

# Lab #7

## Eng:Ali Maarouf

**http://fac.ksu.edu.sa/alimaarouf**

# Objectives

- Learn about Functions that return a value.
- Learn about the importance of argument ordering.
- Learn about functions which return more than one value (call_by_reference).

# Functions in C++

Functions are used to provide modularity to a program. Creating an application using function makes it easier to understand, edit, check errors etc.

## Syntax of Function

```
return-type function-name (parameters)
{
 // function-body
}
```

- **return-type :** suggests what the function will return. It can be int, char, some pointer or even a class object. There can be functions which does not return anything, they are mentioned with **void**.
- **Function Name :** is the name of the function, using the function name it is called.
- **Parameters :** are variables to hold values of arguments passed while function is called. A function may or may not contain parameter list.
- **Function body :** is he part where the code statements are written

# Declaring, Defining and Calling Function

```
#include < iostream>
using namespace std;
int sum (int x, int y);    //declaring function
int main()
{
 int a = 10;
 int b = 20;
 int c = sum (a, b);     //calling function
 cout << c;
}
int sum (int x, int y)  //defining function
{
 return (x + y);
}
```

Function declaration, is done to tell the compiler about the existence of the function. Function's return type, its name & parameter list is mentioned. Function body is written in its definition. Lets understand this with help of an example.

# Calling Of A Function

the function can be called using either of the following methods:

i) call by value.      ii) call by reference.

```
#include <iostream>
using namespace std;
void by_value(int a)
{
a+=10;
}
void by_ref2(int &a){
a+=10;
}
int main(){
 int x=40;
 by_value(x);
 cout<<x<<endl;
 by_ref2(x);
 cout<<x<<endl;
return 0;}
```

## Call By Value

In call by value method, the called function creates its own copies of original values sent to it. Any changes, that are made, occur on the function’s copy of values and are not reflected back to the calling function.

## Call By Reference

In call be reference method, the called function accesses and works with the original values using their references. Any changes, that occur, take place on the original values are reflected back to the calling code.

# Exercise 1:

This program computes the total cost of purchases made, including 5% sales tax, on number items at a cost of price each.

```
#include <iostream>
using namespace std;

double total_cost (int number,  double price); // (1) Function declaration

int main( )
{
        double p, bill;
        int k;

        cout << "Enter the number of items purchased: ";
        cin >> k;
        cout << "Enter the price per item $";
        cin >> p;

        bill = total_cost(k, p);               // (2) Function call
        cout.setf (ios::fixed);
        cout.setf (ios::showpoint);
        cout.precision(2);
        cout << k << " items at "  << "$" << p << " each.\n"
                << "Final bill, including tax, is $" << bill  << endl;

        return 0;
}

// (3) Function definition
double total_cost (int number, double price)         // Function heading
{
        const double TAX_RATE = 0.05;
        double subtotal;
        subtotal = price * number;
        subtotal = subtotal + subtotal*TAX_RATE;
        return (subtotal);
      }
```

# Exercise 2:

Use function to compute the side area and the cross section area of a cylinder .

```
#include<iostream>
#include<cmath>
using namespace std;
const double PI = 3.14159;

double area(double r);  // Function that computes cross section area

double area(double r, double h);  // Function that computes side area
int main()
{
     double h, r, c_area, s_area;  //variables local to the main function

      cout << "Enter the radius and the height of a cylinder ";
      cin >> r >> h;
      cout << endl;
      c_area = area(r);
      cout << "The cross section area of the cylinder is " << c_area <<
endl;
      s_area = area(r, h);
      cout << "The side area of the cylinder is " << s_area <<"  \n\n";

      return 0;
}

double area (double r)
{
    double a;
    a  = (2*PI*pow(r,2));
    return a;
}
double area (double r, double h)
{
      double area;
      area = 2*PI*r*h;
      return area;
}
```

# Exercise 3:

- The function *Process (int a, int b, int c, int& s, double& g)* computes the sum and average of 3 integer numbers. Write a C++ program that calls the function *Process*

```
#include <iostream>
using namespace std;
void  process (int a, int b, int c, int& s, double& g); // (1) Function declaration
int main( )
{
      double  avg;
      int  x1,x2,x3,sum ;
      cout << “ please enter 3 integer numbers \n”;
      cin >> x1 >> x2 >> x3 ;

      process (x1,x2,x3,sum,avg);                   // (2) Function call
      cout<<"sum is: "<<sum<<" avg is: "<<avg;
      return 0;
}
// (3) Function definition
void  process (int a, int b, int c, int& s, double& g)  // Function heading
{
     s = a + b + c;
       g = s/3.0;

}
```

# PostLab 5

Q1.

- Write a separate function to calculate the tax called:

  double taxed_price (double  price, double TAX_RATE)
  And then modify the code of Ex.1.

Q2.

- Modify the function of exercise 3 to be of the form:
  *double  Process (int a, int b, int c, int& s)*